

PATENT  
81940.0074

Express Mail Label No. EV 325 215 270 US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Koji SONODA et al.

Serial No: Not assigned

Filed: April 2, 2004

For: FILE SYSTEM

Art Unit: Not assigned

Examiner: Not assigned

TRANSMITTAL OF PRIORITY DOCUMENT

Mail Stop PATENT APPLICATION

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Dear Sir:

Enclosed herewith is a certified copy of Japanese patent application No. 2003-383249 which was filed November 13, 2003, from which priority is claimed under 35 U.S.C. § 119 and Rule 55.

Acknowledgment of the priority document(s) is respectfully requested to ensure that the subject information appears on the printed patent.

Respectfully submitted,

HOGAN & HARTSON L.L.P.

Date: April 2, 2004

By: 

Anthony J. Orler

Registration No. 41,232

Attorney for Applicant(s)

500 South Grand Avenue, Suite 1900

Los Angeles, California 90071

Telephone: 213-337-6700

Facsimile: 213-337-6701

日本国特許庁  
JAPAN PATENT OFFICE

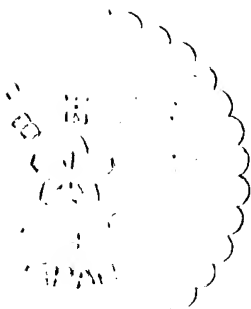
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年11月13日  
Date of Application:

出願番号 特願2003-383249  
Application Number:  
[ST. 10/C]: [JP 2003-383249]

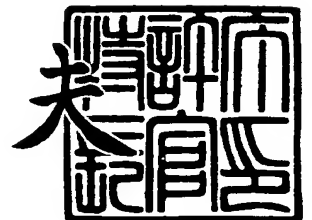
出願人 株式会社日立製作所  
Applicant(s):



2004年 2月12日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



出証番号 出証特2004-3008529

【書類名】 特許願  
【整理番号】 K03008791A  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 12/00  
【発明者】  
    【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所  
                                システム開発研究所内  
    【氏名】 藺田 浩二  
【発明者】  
    【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所  
                                システム開発研究所内  
    【氏名】 岩崎 正明  
【発明者】  
    【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所  
                                ソフトウェア事業部内  
    【氏名】 杵島 伸一  
【特許出願人】  
    【識別番号】 000005108  
    【氏名又は名称】 株式会社 日立製作所  
【代理人】  
    【識別番号】 100075096  
    【弁理士】  
    【氏名又は名称】 作田 康夫  
【選任した代理人】  
    【識別番号】 100100310  
    【弁理士】  
    【氏名又は名称】 井上 学  
【手数料の表示】  
    【予納台帳番号】 013088  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1

**【書類名】 特許請求の範囲****【請求項 1】**

クライアント計算機からファイルへのアクセス要求を受信する第一の計算機と、  
前記第一の計算機に接続され、ファイルの管理情報を格納する第一の記憶装置システムと、

前記第一の計算機からデータへのアクセス要求を受信する第二の計算機と、  
前記第二の計算機に接続され、ファイルデータを格納する第二の記憶装置システムと、  
前記クライアント計算機、前記第一の計算機、及び前記第二の計算機に接続されるネットワークとを有する計算機システムであって、

前記第一の計算機は、クライアント計算機からファイルデータを受信する毎に、ファイルデータに識別情報を割り当てて、ファイルデータを前記第二の計算機を介して前記第二の記憶装置システムに格納し、

前記第一の記憶装置システムは、前記第一の計算機がファイルデータに割り当てた識別情報と、クライアントから指定される当該ファイルデータを有するファイルのファイル名とを記憶しており、

前記第一の計算機は、クライアント計算機からファイルへのライト要求を受信すると、  
前記第二の記憶装置システムに既に格納されているライト対象のファイルのファイルデータに割り当てられている第一の識別情報とは異なる第二の識別情報を、前記ライト要求に伴い前記クライアントから受信するライトデータに割り当て、

前記第二の計算機を介して、前記ライトデータを、前記第二の記憶装置システムに既に格納されている前記ライト対象のファイルデータとは異なる前記第二の記憶装置システム内の記憶領域に格納し、

前記第二の識別情報を、前記ライト対象のファイルのファイル名と前記第一の識別情報とに対応付けて、前記第一の記憶装置システムに格納することを特徴とする計算機システム。

**【請求項 2】**

請求項 1 記載の計算機システムにおいて、

前記第二の記憶装置システムは、ファイルデータを格納するためのファイルコンテナを有しており、

前記ライトデータは、前記ライト対象のファイルのファイルデータが格納されているファイルコンテナとは異なるファイルコンテナに格納されることを特徴とする計算機システム。

**【請求項 3】**

請求項 1 記載の計算機システムにおいて、

更にクライアント計算機からのファイルへのアクセス要求を受信し、受信したアクセス要求を前記第一の計算機が使用しているプロトコルに従ったアクセス要求に変換して、変換後のアクセス要求を前記第一の計算機に送信する第三の計算機を有することを特徴とする計算機システム。

**【請求項 4】**

請求項 1 記載の計算機システムにおいて、

前記第二の記憶装置システムは更に、少なくとも一以上のファイルのファイル名とファイルの識別情報との組を有するビューデータを格納しており、

前記第一の記憶装置システムは更に、前記ビューデータの格納位置を含むビューの管理情報を格納しており、

前記第一の計算機は、クライアント計算機からビューデータの読み出し要求を受信した場合に、前記第一の記憶装置システムに格納されているビューの管理情報に基づいて、前記第二の計算機を介して前記第二の計算機からビューデータを読み出すことを特徴とする計算機システム。

**【請求項 5】**

請求項 4 記載の計算機システムにおいて、

前記第二の記憶装置システムは更に、ビューデータと時刻情報とを関連付けて格納しており、

ビューデータは、当該ビューデータに関連付けられた時刻情報が示す時刻において、前記第二の記憶装置システムに格納されているファイルデータに対応するファイル名とファイルの識別情報との組を有することを特徴とする計算機システム。

【請求項 6】

請求項 5 記載の計算機システムにおいて、

前記第一の計算機は、クライアント計算機から新規ファイルの作成要求を受信した場合に、作成された新規ファイルのファイル名とファイルの識別情報の組と、前記第二の記憶装置システムに格納されている他のファイルのファイル名とファイルの識別情報の組を有するビューデータを、前記新規ファイルが作成された時刻を示す時刻情報と関連付けて、前記第二の計算機を介して前記第二の記憶装置システムに格納することを特徴とする計算機システム。

【請求項 7】

請求項 5 記載の計算機システムにおいて、

前記第一の計算機は、前記ライト要求を受信した場合に、前記第一の識別情報と前記ファイル名との組、及び前記第二の識別情報と前記ファイル名との組を含むビューデータを、前記ライトデータを前記第二の記憶装置システムに書き込んだ時刻を示す時刻情報と共に、前記第二の計算機を介して前記第二の記憶装置システムに格納することを特徴とする計算機システム。

【請求項 8】

請求項 7 記載の計算機システムにおいて、

前記第一の計算機は、クライアント計算機から時刻情報を含むビューデータ読み出し要求を受信した場合に、ビューデータと関連付けられている時刻情報のうち、ビューデータ読み出し要求に含まれる時刻情報より古い時刻情報の中で最も新しい時刻情報を選択し、選択された時刻情報と関連付けられているビューデータを前記第二の計算機を介して前記第二の記憶装置システムから読み出し、前記クライアント計算機に送信することを特徴とする計算機システム。

【請求項 9】

請求項 1 記載の計算機システムにおいて、

更に、クライアント計算機のファイルに対するアクセス権限をチェックする第三の記憶装置システムを有しており、

前記第一の計算機は、クライアント計算機からファイルへのアクセス要求を受信した場合に、前記第三の計算機にアクセス権限チェック要求を送信し、前記第三の計算機から受信するアクセス権限のチェック結果に応じて、クライアント計算機からのファイルへのアクセス要求に従って、ファイルへのアクセス処理を実行するか否かを決定することを特徴とする計算機システム。

【請求項 10】

クライアント計算機から受信するファイルへのアクセス要求に従って、ファイルへのアクセス処理を実行するためのプログラムであって、

前記プログラムは、

クライアント計算機からファイルへのアクセス要求を受信する第一の計算機と、

前記第一の計算機に接続され、ファイルの管理情報を格納する第一の記憶装置システムと、

前記第一の計算機からデータへのアクセス要求を受信する第二の計算機と、

前記第二の計算機に接続され、ファイルデータを格納する第二の記憶装置システムと、

前記クライアント計算機、前記第一の計算機、及び前記第二の計算機に接続されるネットワークとを有する計算機システムにおいて実行され、

前記第一の計算機によって、ファイルデータをクライアント計算機から受信する度に当該ファイルデータに識別情報を割り当てるためのコードと、

前記第二の計算機によって、前記第一の計算機がクライアント計算機から受信したファイルデータを前記第二の記憶装置システムに格納するためのコードと、

前記第一の計算機によって、第一の計算機がファイルデータに割り当てた識別情報と、クライアントから指定される当該ファイルデータを有するファイルのファイル名とを前記第一の記憶装置システムに格納するためのコードと、

前記第一の計算機によって、クライアント計算機からファイルへのライト要求を受信するためのコードと、

前記第一の計算機によって、前記第二の記憶装置システムに格納されているライト対象のファイルのファイルデータに割り当てられている第一の識別情報とは異なる第二の識別情報を、前記ライト要求に伴い前記クライアントから受信するライトデータに割り当てるためのコードと、

前記第二の計算機によって、前記ライトデータを前記第二の記憶装置システムに格納するためのコードと、

前記第一の計算機によって、前記第二の識別情報を、前記ライト対象のファイルのファイル名と前記第一の識別情報とに対応付けて、前記第一の記憶装置システムに格納するためのコードとを有することを特徴とするプログラム。

【請求項 11】

請求項 10 記載のプログラムにおいて、

前記第二の記憶装置システムは、ファイルデータを格納するためのファイルコンテナを有しており、

前記ライトデータを前記第二の記憶装置システムに格納するためのコードは、前記ライトデータを、前記ライト対象のファイルのファイルデータが格納されているファイルコンテナとは異なるファイルコンテナに格納するためのコードを有することを特徴とするプログラム。

【請求項 12】

請求項 10 記載のプログラムにおいて、

更に前記計算機システムは、前記ネットワークに接続される第三の計算機を有しており、

前記プログラムは、前記第三の計算機によって、クライアント計算機からのファイルへのアクセス要求を受信し、受信したアクセス要求を前記第一の計算機が使用しているプロトコルに従ったアクセス要求に変換して、変換後のアクセス要求を前記第一の計算機に送信するためのコードを有することを特徴とするプログラム。

【請求項 13】

請求項 10 記載のプログラムにおいて、

更に、前記第二の記憶装置システムによって、少なくとも一以上のファイルのファイル名とファイルの識別情報との組を有するビューデータを格納するためのコードと、

前記第一の記憶装置システムによって、前記ビューデータの格納位置を含むビューの管理情報を格納するためのコードと、

前記第一の計算機によって、クライアント計算機からビューデータの読み出し要求を受信するためのコードと、

前記第一の計算機によって、ビューデータの読み出し要求に従って、前記第一の記憶装置システムに格納されているビューの管理情報に基づいて、前記第二の計算機を介して前記第二の計算機からビューデータを読み出すためのコードとを有することを特徴とするプログラム。

【請求項 14】

請求項 13 記載のプログラムにおいて、

前記第二の記憶装置システムによって、ビューデータを格納するためのコードは、ビューデータと時刻情報とを関連付けて格納するためのコードを有しており、

ビューデータは、当該ビューデータに関連付けられた時刻情報が示す時刻において、前記第二の記憶装置システムに格納されているファイルデータに対応するファイル名とファ

イルの識別情報との組を有することを特徴とするプログラム。

【請求項 15】

請求項 14 記載のプログラムにおいて、更に、

クライアント計算機から新規ファイルの作成要求を受信した場合に、前記第一の計算機によって、作成された新規ファイルのファイル名とファイルの識別情報の組、及び前記第二の記憶装置システムに格納されている他のファイルのファイル名とファイルの識別情報の組を有するビューデータを、前記新規ファイルが作成された時刻を示す時刻情報と関連付けて、前記第二の計算機を介して前記第二の記憶装置システムに格納するためのコードを有することを特徴とするプログラム。

【請求項 16】

請求項 14 記載のプログラムにおいて、更に、

前記ライト要求を受信した場合に、前記第一の計算機によって、前記第一の識別情報と前記ファイル名との組、及び前記第二の識別情報と前記ファイル名との組を含むビューデータを、前記ライトデータを前記第二の記憶装置システムに書き込んだ時刻を示す時刻情報と共に、前記第二の計算機を介して前記第二の記憶装置システムに格納するためのコードを有することを特徴とするプログラム。

【請求項 17】

請求項 16 記載のプログラムにおいて、更に、

クライアント計算機から時刻情報を含むビューデータ読み出し要求を受信した場合に、前記第一の計算機によって、ビューデータと関連付けられている時刻情報のうち、ビューデータ読み出し要求に含まれる時刻情報より古い時刻情報の中で、最も新しい時刻情報を選択し、選択された時刻情報と関連付けられているビューデータを前記第二の計算機を介して前記第二の記憶装置システムから読み出し、前記クライアント計算機に送信するためのコードを有することを特徴とするプログラム。

【請求項 18】

請求項 1 記載のプログラムにおいて、更に

前記計算機システムは、クライアント計算機のファイルに対するアクセス権限をチェックする第三の記憶装置システムを有しており、

クライアント計算機からファイルへのアクセス要求を受信した場合に、前記第一の計算機によって、前記第三の計算機にアクセス権限チェック要求を送信し、前記第三の計算機から受信するアクセス権限のチェック結果に応じて、クライアント計算機からのファイルへのアクセス要求に従ってファイルへのアクセス処理を実行するか否かを決定するためのコードを有することを特徴とするプログラム。

【請求項 19】

計算機システムにおけるファイルへのアクセス方法であって、

前記計算機システムは、

クライアント計算機からファイルへのアクセス要求を受信する第一の計算機と、

前記第一の計算機に接続され、ファイルの管理情報を格納する第一の記憶装置システムと、

前記第一の計算機からデータへのアクセス要求を受信する第二の計算機と、

前記第二の計算機に接続され、ファイルデータを格納する第二の記憶装置システムと、

前記クライアント計算機、前記第一の計算機、及び前記第二の計算機に接続されるネットワークとを有しており、

前記第二の記憶装置システムは、ファイルデータを格納するためのファイルコンテナを有しており、

前記第一の計算機が、クライアント計算機からファイルへのライト要求を受信するステップと、

前記第二の記憶装置システムに格納されているライト対象のファイルのファイルデータに前記第一の計算機が割り当てた第一の識別情報とは異なる第二の識別情報を、前記ライト要求に伴い前記クライアント計算機から受信するライトデータに前記第一の計算機が割

り当てるステップと、

前記第二の計算機がライトデータを、前記ライト対象のファイルのファイルデータが格納されているファイルコンテナとは異なるファイルコンテナに格納するステップと、

前記第一の計算機が、前記第二の識別情報を、前記ライト対象のファイルのクライアント計算機から指定されたファイル名と前記第一の識別情報と対応付けて前記第一の記憶装置システムに格納するステップとを有することを特徴とするファイルへのアクセス方法。

【請求項 2 0】

請求項 1 9 記載のファイルへのアクセス方法であって、

前記計算機システムは、更にプロトコル変換を実行する第三の計算機を有しており、

更に、クライアント計算機からのファイルへのアクセス要求を前記第三の計算機が受信するステップと、

前記第三の計算機が、受信したアクセス要求を前記第一の計算機が使用しているプロトコルに従ったアクセス要求に変換して、変換後のアクセス要求を前記第一の計算機に送信するステップとを有することを特徴とするファイルへのアクセス方法。



【書類名】明細書

【発明の名称】ファイルシステム

【技術分野】

【0001】

本発明は、計算機からアクセスされるファイルを格納したファイルシステムに関し、特にネットワークに接続された複数のサーバ計算機及び複数の記憶装置によって実現される分散ファイルシステムに関する。

【背景技術】

【0002】

近年、IT (Information Technology) の進化に伴い、さまざまな業務が電子化され、従来紙を用いて処理されていた業務は、電子的なデータである電子ファイル（以下、単に「ファイル」とも言う。）を用いて処理されるようになってきている。一方、契約書や取り引き履歴のように、監査や裁判の証拠として使用される書類は、その書類がオリジナルであり、かつ改ざんされていないことが保障されていなければならない。

【0003】

現在、電子ファイルを格納するための記憶装置としては磁気ディスクが主流であり、計算機が磁気ディスク上にファイルシステムを構築し、計算機はファイルシステム上にファイルとして電子データを格納する形態が一般的である。磁気ディスクとファイルシステムを用いたファイルの格納方式では、ファイルを変更したり消去したりすることは容易であるため、ファイルがオリジナルであり、かつ改竄されていないことを保証するのは困難であった。

【0004】

この問題を解決する技術として、例えば特許文献1に挙げられている技術が存在する。この技術では、記憶装置が当該記憶装置に格納するデータ全体からハッシュ値を計算し、算出したハッシュ値をデータの識別子とする。データが異なればハッシュ値も異なるため、ユーザが同じハッシュ値を用いてデータにアクセスする限り、当該データには変更がないことが保障される。

【0005】

【特許文献1】国際公開第99/38092号パンフレット

【発明の開示】

【発明が解決しようとする課題】

【0006】

特許文献1では、データが変更されていないことを保障することは可能だが、ユーザがデータを取得するためには、当該データに対応するハッシュ値をユーザが記憶装置に対して与える必要がある。従って、保存したデータとハッシュ値の対応をユーザが管理する必要があり、管理コストが大きいという課題がある。

【0007】

また特許文献1では、ユーザがデータに任意の名前をつけて当該データを記憶装置に保存することができないため、ユーザはハッシュ値を紛失すると、当該データにアクセスできなくなってしまう。

【0008】

さらに特許文献1では、変更されたデータは元のデータとは全く別のデータとして、記憶装置に登録され管理され、変更後のデータと元のデータとの関係は記憶装置では管理していない。このため、データの変更後に、変更前のデータにデータを復元したい場合には、ユーザが変更前のデータと変更後のデータとの間の関係を管理しておく必要がある。

【0009】

そこで、クライアント計算機からアクセスされるファイルを格納する計算機システムであって、クライアント計算機からの要求に基づいて記憶装置に格納されるデータ各々に異なる識別情報を付与して一旦格納されたデータは更新されることのないようデータを管理

し、当該識別情報とクライアント計算機から指定されるファイル名とを関連付けて管理することのできる計算機システムを開示する。

【0010】

また、クライアントからライト要求を受信した場合に、更新前のデータと更新後のデータとを関連付けて管理することのできる計算機システムを開示する。

【課題を解決するための手段】

【0011】

計算機システムは、クライアント計算機からファイルへのアクセス要求を受信する第一の計算機と、第一の計算機に接続され、ファイルの管理情報を格納する第一の記憶装置システムと、第一の計算機からデータへのアクセス要求を受信する第二の計算機と、第二の計算機に接続され、ファイルデータを格納する第二の記憶装置システムとを有する。

【0012】

第一の計算機は、クライアントからファイルに含まれるファイルデータを受信する毎に、ファイルデータに識別情報を割り当てて、ファイルデータを第二の計算機を介して第二の記憶装置システムに格納する。第一の記憶装置システムは、第一の計算機がファイルデータに割り当てた識別情報と、クライアントから指定されるファイル名とを記憶している。

【0013】

第一の計算機は、クライアント計算機からファイルへのライト要求を受信すると、第二の記憶装置システムに既に格納されているライト対象のファイルのファイルデータに割り当てられている第一の識別情報とは異なる第二の識別情報を、ライト要求に伴いクライアントから受信するライトデータに割り当てる。そして第一の計算機は、第二の計算機を介して、既に第二の記憶装置システムに格納されているライト対象のファイルのファイルデータとは異なる第二の記憶装置システム内の記憶領域に、ライトデータを格納する。

【0014】

更に第一の計算機は、第二の識別情報を、ライト対象のファイルのファイル名と第一の識別情報とに対応付けて、第一の記憶装置システムに格納する。

【発明の効果】

【0015】

クライアント計算機から受信したライトデータに各々異なる識別情報を付与し、一旦格納されたデータは更新されることのないよう管理すると共に、ライトデータに付与された識別情報とクライアント計算機から指定されるファイル名とを関連付けて管理する計算機システムが実現できる。

【0016】

また、更新前のデータと更新後のデータとを関連付けて管理することのできる計算機システムが実現できる。

【発明を実施するための最良の形態】

【0017】

以下、図を用いて本発明の実施形態を説明する。尚、以下に説明する実施形態により、本発明が限定されるものではない。

【実施例1】

【0018】

図1は本発明におけるファイルシステムの概念の一例を示す図である。

【0019】

本発明のファイルシステムは、ファイル120と、ファイル120やView100が登録されるView100とで構成される。

【0020】

View100は図1に示す例のように、階層構造を有しており、例えば図1のView3にはView4が登録されている（以下、図1のView3及びView4の様な階層関係を、View3はView4の親View、View4はView3の子Viewである、と表現する。）。

## 【0021】

各View100はViewデータ110を有しており、Viewデータ110にはファイル120の他に子View100が登録されている。Viewデータ110は、ファイル120やView110が作成されたり、消去されるごとに新たに作成され、当該Viewデータに登録されるファイルやViewの識別情報を有する。Viewデータは、当該Viewデータが作成された時刻情報と共にView内で管理される。従って、Viewデータと当該Viewデータと共に管理されている時刻情報を参照すれば、ファイル120やViewの作成、削除等の更新履歴を知ることができる。

## 【0022】

例えば図1の例では、View4(100A)にViewデータ110AとViewデータ110Bが登録されている。Viewデータ110Aは時刻date1におけるView4の状態を示す情報を保持しており、Viewデータ110Aにはファイル120Aのみが登録されている。つまりViewデータ110Aは、View4には時刻date1時点においてファイル120Aのみが属していたことを示している。Viewデータ110Bは時刻date2におけるView4の状態を示す情報を保持しており、Viewデータ110Bにはファイル120A、120Bの2つのファイル及びView100Dが登録されている。つまり、Viewデータ110Bは、View4にはdate2の時点においてファイル120Aとファイル120Bの2つのファイル及びView6が属していたことを示している。

## 【0023】

また、View100BにはViewデータ100Cが登録されている。Viewデータ100Cは時刻date1におけるView3の状態を示す情報であり、Viewデータ100CにはView4のみが登録されている。つまりViewデータ100Cは、View3には時刻date1において子ViewであるView4のみが属していたことを示している。またView100BにはViewデータ100Cの他に登録されているViewデータは存在しないから、時刻date1以降View100Bの状態に変化がないことも分かる。

## 【0024】

このように、Viewの登録情報に変更がある度にViewデータを作成することで、任意の時点におけるファイルおよびViewのセットをファイルシステム内で管理することができる。

## 【0025】

図2は、本発明を適用した計算機システムの一例を示す図である。本計算機システムは、ユーザが使用する複数の計算機（以下クライアントと呼ぶ。）1000、ファイル管理情報（以下メタデータと呼ぶ。）を管理するサーバ計算機であるコントロールノード（以下CNと呼ぶ。）1200、CN1200と接続されメタデータを格納する記憶装置1230A、ファイルデータやViewデータを管理するサーバ計算機であるストレージノード（以下SNと呼ぶ。）1300、SN1300と接続されファイルデータやViewデータを格納する記憶装置1230B、NFSプロトコルを用いてファイルアクセスを行う計算機（以下、NFSクライアントと呼ぶ。）1100、CIFSプロトコルを用いてファイルアクセスを行う計算機（以下CIFSクライアントと呼ぶ。）1110、NFSプロトコルやCIFSプロトコルに従ったファイルアクセス要求をCN1200が用いているファイルアクセスプロトコルに変換する計算機（以下レガシーゲートウェイと呼ぶ。）1600、レガシーゲートウェイ1600に接続されレガシーゲートウェイがNFSクライアント1100やCIFSクライアント1110から受信したファイルデータを一時的に格納する記憶装置1230C、CN1200やSN1300と連携するサーバ計算機（以下、外部ノードと呼ぶ。）1500、外部ノード1500に接続され外部ノードが使用するデータを格納する記憶装置1230D、および本計算機システムを管理する計算機（以下管理ノードと呼ぶ。）1400を有する。

## 【0026】

複数のクライアント1000、CN1200、SN1300、NFSクライアント1100、CIFSクライアント1110、レガシーゲートウェイ1600、外部ノード15

00、管理ノード1400は、ネットワーク1700によって接続され、これらの計算機は相互に通信可能である。

【0027】

図13はSN1300及び記憶装置1230Bの一例を示す図である。

【0028】

SNはプロセッサ、プロセッサによって実行されるストレージサーバプログラム（以下S-SVR1330と呼ぶ。）を格納したメインメモリ、入出力部、ネットワークに接続するためのネットワークインタフェース、及び記憶装置に接続するためのディスクアダプタを有する計算機である。ここでS-SVR1330は、CN1200がNFSやCIFSの様な一般的なファイルアクセスプロトコルを用いて、記憶装置1230Bに格納されているファイルデータやViewデータにアクセスできるよう、制御するためのプログラムである。

【0029】

尚、クライアント1000、CN1200、NFSクライアント1100、CIFSクライアント、外部ノード1500、管理ノードも、図13に示すSN1300と同様の構成を有しているが、メインメモリに格納されるプログラムやデータが、SN1300とは異なる。また、クライアント1000、NFSクライアント1100、CIFSクライアント1110、管理ノード1400はディスクアダプタを有していない点も、SN1300とは異なる。

【0030】

記憶装置1230Bは、計算機と接続するためのアダプタ、計算機から受信したデータやディスクから読み出したデータを一時的に格納するキャッシュメモリ、メインメモリ、メインメモリに格納されたプログラムやデータを用いてディスクを制御するディスク制御プロセッサ、ディスクに接続するためのディスクアダプタ、及び一または複数のディスクを有する。ディスクには、ファイルデータを格納しているファイルコンテナ1310及びViewデータを格納しているViewコンテナ1320が格納されている。

【0031】

尚、記憶装置1230A、記憶装置1230C、記憶装置1230Dも図13に示す記憶装置1230Bと同様の構成を有しているが、ディスクに格納されているデータは記憶装置1230Bとは異なる。

【0032】

更に、SN1300と記憶装置1230Bは、NAS (Network Attached storage)として1の筐体内に構成された1の装置であっても良い。同様に、CN1200と記憶装置1230A、レガシーゲートウェイ1600と記憶装置1230C、及び外部ノード1500と記憶装置1230Dも各々NASとして一筐体内に構成されていても良い。

【0033】

図2に戻って計算機システムの説明を続ける。

【0034】

CN1200は、クライアント1000やレガシーゲートウェイと通信する時の接続情報（以下セッション情報と呼ぶ。）を保持するセッションテーブル（以下STと呼ぶ。）1210、オープンしたファイルの情報を保持するオープンファイルテーブル（以下OFTと呼ぶ。）1260、ファイルの管理を行うプログラムであるコントロールサーバ（以下CSVRと呼ぶ。）1270、記憶装置1230A上のデータベースを管理するデータベース管理サーバプログラム（以下DBMSと呼ぶ。）1220をメインメモリ内に格納している。

【0035】

記憶装置1230Aは、ファイルやViewの属性情報を格納するデータベースである属性テーブル（以下ATと呼ぶ。）1240、ファイルやViewの位置情報を格納するデータベースであるロケーションテーブル（以下LTと呼ぶ。）1250をディスクに格納している。

【0036】

外部ノード1500は、ファイルシステムのアクセス履歴を管理するプログラムである

履歴マネージャ 1 5 1 0、ユーザへの課金情報を管理するプログラムである課金マネージャ 1 5 2 0、アクセス権限情報を管理するプログラムである ACL マネージャ 1 5 3 0、ファイルの検索を行なう検索プログラム 1 5 4 0 をメインメモリに格納している。図 2 に示す例では、これらのプログラムは、一つの外部ノード 1 5 0 0 に搭載されているが、各プログラムがそれぞれ異なる計算機に搭載されていてもよい。

#### 【0 0 3 7】

記憶装置 1 2 3 0 D は、外部ノード 1 5 0 0 上の履歴マネージャ 1 5 1 0、課金マネージャ 1 5 2 0、ACL マネージャ 1 5 3 0 のそれぞれが使用するデータベースをディスク内に格納する。

#### 【0 0 3 8】

図 3 は、本実施形態におけるファイルの管理方法を説明する図である。ファイルは、ファイルの属性情報を保持するファイル属性エントリ 2 0 0 0 と、ファイルの実体でありユーザによって格納されるファイルデータを格納するファイルコンテナ 1 3 1 0 で構成されている。

#### 【0 0 3 9】

ファイル属性エントリ 2 0 0 0 は記憶装置 1 2 3 0 A の AT 1 2 4 0 に保持され、ファイルコンテナ 1 3 1 0 は記憶装置 1 2 3 0 B に保持される。

#### 【0 0 4 0】

ファイル属性エントリ 2 0 0 0 は、C-SVR1270 によってファイルデータに付与される計算機システム内で一意な識別子である GUID、ユーザがファイルに付与する文字列名である name、ファイルの作成日付である C d a t e、ファイルの更新日付である M d a t e、ファイルコンテナ 1 3 1 0 に格納されているファイルデータのサイズを示す size、ファイルの操作種別を示す t y p e、更新前のファイルの元データの識別子である a n c \_ G U I D、ファイルコンテナの格納位置を示すロケーション情報である c o n t e n t s \_ L o c、c o n t e n t s \_ L o c で示されるファイルコンテナがファイルのどのオフセットのデータを保持しているのかを示す c o n t e n t s \_ I n f o、ファイルに対するアクセス制御情報を保持するサーバの位置情報である A C T \_ M G R \_ I D、ファイルに対する課金情報を保持するサーバの位置情報である A C T \_ M G R \_ I D で構成される。

#### 【0 0 4 1】

ここでファイルの操作種別には、WORM 型と追記型の 2 種類がある。WORM 型ファイルは、一旦ファイルコンテナ 1300 にファイルデータが書き込まれ当該ファイルコンテナがファイル属性エントリ 2000 に登録されると、当該ファイルコンテナにデータを追記することができなくなるファイルである。追記型ファイルは、ファイルデータが書き込まれたファイルコンテナ 1300 がファイル属性エントリ 2000 に登録された後も、当該ファイルコンテナ 1300 にデータを追記できるファイルである。但し追記型ファイルにおいても、すでに書き込み済みのデータを上書きすることはできない。本実施形態においては、ファイルの操作種別は WORM 型であるものとする。

#### 【0 0 4 2】

図 3 の例は、ファイル属性エントリ 2 0 0 0 B とファイルコンテナ 1 3 1 0 B で構成されるファイル（以下ファイル B と呼ぶ。）と、ファイル B の情報の一部を変更して作成されたファイル（以下ファイル A と呼ぶ）との関係を示している。

#### 【0 0 4 3】

ファイル B は 5 1 2 M B のデータをファイルコンテナ 1 3 1 0 B に保持している。

#### 【0 0 4 4】

ファイル A はファイルコンテナ 1310B に保持されたファイル B のファイルデータのうち、オフセット 0 からオフセット 2 0 0 M までに対応するが変更されているファイルである。ファイル A はファイル属性エントリ 2 0 0 0 A 及びファイル属性エントリ 2 0 0 0 B とファイルコンテナ 1 3 1 0 A 及びファイルコンテナ 1 3 1 0 B とで構成される。ファイル属性エントリ 2 0 0 0 A の GUID はファイル A に対して C-SVR が付与した識別子であり、ファイル B の GUID とは異なる値を持つ。ファイル属性エントリ 2 0 0 0 A の c o n t e n t s \_

Loc は、更新されたデータ即ちファイル B とファイル A の差分内容を保持するファイルコンテナ 1310A の格納位置を示しており、ファイル属性エントリ 2000A の contents\_Inf o は、更新されたデータがファイル A のどの部分に相当するデータであることを示すオフセットを示している。また、ファイル属性エントリ 2000A の anc \_G U I D は、ファイル B の G U I D を保持している。従ってファイル属性エントリ 2000A の anc \_G U I D を用いれば、クライアント計算機からファイル A に対するアクセスがあった場合に、ファイルコンテナ 1310A が保持していないデータをファイル B のファイルコンテナを参照して取得することが可能となる。

#### 【0045】

図 4 は V i e w の管理方法の一例を示している。V i e w には、ファイルや子 V i e w のユーザから指定された文字列名とそのファイル若しくは子 View の識別子である G U I D のペアを複数登録することができ、View を用いることによって、ユーザは文字列名を指定してファイルを特定することができるようになる。また、外部ノード 1500 上の検索エンジン 1540 と View とを連携させ、検索エンジンが実行されることによって得られた結果を View に登録することができる。また、検索方法をスクリプトファイルとして記憶装置 1230D に格納しておき、定期的に当該スクリプトファイルを検索エンジン 1540 が実行して、その検索結果を View に登録することで、動的な V i e w の更新を行うことができる。

#### 【0046】

V i e w は V i e w 属性エントリ 2100 と V i e w コンテナ 2200 で構成される。

#### 【0047】

V i e w 属性エントリは、C-SVR1270 によって View に付与される計算機システム内で一意な識別子である G U I D 、ユーザが View に付与する文字列名である name、View の作成日付である C d a t e、View の更新日付である M d a t e、View 属性であることを示す t y p e、親 View の GUID である parent\_ G U I D、View コンテナの格納位置を示す c o n t e n t s \_L o c、View コンテナ内に格納される最新の View デリミタの View コンテナの先頭位置からのオフセットである contents\_ i n f o、View に対するアクセス制御情報を保持するサーバを示す A C T \_M G R \_I D、View に対する課金情報を保持するサーバを示す A C T \_M G R \_I D に加え、スクリプトファイルの G U I D である s c r i p t \_G U I D、スクリプトの実行間隔を示す s c r i p t \_i n t e r v a l を有する。View 属性エントリ 2100 は、ファイル属性エントリ同様、記憶装置 1230A の A T 1240 に格納される。このため、t y p e の値により V i e w 属性かファイル属性かを判断する。前述の通り、ファイル属性の場合には、W O R M 型か追記型のいずれかの値が t y p e に格納される。V i e w 属性の場合は、V i e w 型を示す値が格納される。

#### 【0048】

V i e w コンテナ 2200 は、複数の V i e w デリミタ 2210 と複数の V i e w データ 2220 で構成される。

#### 【0049】

V i e w デリミタ 2210 は V i e w データ 2220 とペアで V i e w コンテナ 2200 に登録され、V i e w データ 2220 の登録時刻 d a t e、V i e w データのサイズ s i z e、一つ前の V i e w データの V i e w コンテナ先頭からのオフセット P r e v で構成される。V i e w データ 2220 はファイルの文字列名（以下ファイル名ともいう。）とファイルの G U I D のペアあるいは、V i e w の文字列名（以下 View 名ともいう。）と V i e w の G U I D のペアを複数保持する。

#### 【0050】

ある任意の時刻におけるファイルや View を取得するには、CN1200 の C-SVR1270 が V i e w デリミタ 2210 に登録されている時刻を参照し、当該任意の時刻以前の時刻であって当該任意の時刻に最も近い時刻が登録されている View デリミタ 2210 に対応する V i e w データ 2220 を取得し、この View データ 2220 に登録されている GUID を用いてファイル属性エントリ 2000 や View 属性エントリ 2100 を検索すれば良い。検索の結果、ファイルコン

テナ1310若しくはViewコンテナ2200の格納位置が分かれば、C-SVR1270は当該任意の時刻におけるファイルやViewのデータを取得することができる。

#### 【0051】

図4に示すView属性エントリ2100Aは図1のView4のView属性エントリを、Viewコンテナ2200Aは図1のView4のViewコンテナを、View属性エントリ2100Bは図1のView4の子ViewであるView6のView属性エントリを示している。

#### 【0052】

図4の例では、Viewデリミタ2210AとViewデータ2220AがViewコンテナ2200Aに登録された時点(date1時点)では、View4にはFile1が登録されていたことを示している。このView4に対しdate2時点までにFile2とView6を登録すると、Viewデータ2220BとViewデリミタ2210BがViewコンテナ2200Aに追加される。また、View属性エントリ2100Aのcontents\_infoには、Viewデリミタ2210BのViewコンテナ先頭からのオフセットが設定される。従ってC-SVRはView属性エントリのcontents\_infoとcontents\_Locを用いれば、最新のViewデータが参照できる。

#### 【0053】

図4で述べたView属性エントリ2100は、記憶装置1230AのAT1240に格納されており、Viewコンテナ2200は記憶装置1230Bに格納されている。

#### 【0054】

図5はオープンファイルテーブル（以下OFTとも呼ぶ。）1260の構造の一例を示している。OFT1260は、現在オープン中のファイル若しくはViewに関するデータが登録されるテーブルである。ユーザがクライアント1000、NFSクライアント1100、若しくはCIFSクライアント1110を介してCN1200に対し、ファイル若しくはViewの作成要求を送信すると、当該作成要求に基づきオープンされたファイルやViewの情報がOFT1260に保持される。

#### 【0055】

OFT1260には、ファイル若しくはViewをユーザが指定する際に用いられるハンドル2300、オープンされたファイル若しくはViewの識別子であるGUID2310、当該ファイルやViewにアクセス可能なユーザの証明情報（以下CRED）2320、セッションテーブルのエントリへのポインタ2330、及び当該ファイルやViewを登録している親ViewのGUID（以下親GUID）2340を有している。

#### 【0056】

図6は、セッションテーブル（以下STともいう。）1210の一例を示している。ST1210はクライアント1000とCN1200間のユーザ接続情報を保持するテーブルである。クライアントが最初にCNにアクセスする際、クライアントとCNとの間では通信のためのセッションが確立される。セッション確立の際にはクライアントのユーザ認証も実行される。そこでクライアントとCN間でセッションが確立された際に、STには、クライアントとCN1200間の接続を識別するためのセッション番号2400、接続に使用されるソケットの識別情報2410、接続しているユーザの証明情報（以下CRED）2420が登録される。STを用いれば、CNがクライアントから各種要求を受信した場合に、CNはどのセッションを介して受信した要求かを判別することにより、要求を発行したユーザを特定できるため、ファイルアクセス時のアクセス権チェックやファイル作成時のファイル作成ユーザの特定にSTを利用することができる。

#### 【0057】

以上で説明した計算機システムにおいて、CN1200とSN1300が、クライアント1000やレガシーゲートウェイ1600に対して、ファイルの更新情報を保持するファイルシステムを提供する。尚、レガシーゲートウェイ1600は、NFSクライアント1100やCIFSクライアント1110に対し、NFSプロトコルまたはCIFSプロトコル等のレガシープロトコルを用いてファイルシステムにアクセスするための機能を提供する。即ち、レガシーゲートウェイは、NFSクライアント1100やCIFSクライ

アント 1 1 1 0 から受信した、N F S プロトコル若しくは C I F S プロトコルに従う要求を、C N 1 2 0 0 が処理可能なプロトコルに従う要求に変換して、C N 1 2 0 0 に送信し、また C N 1 2 0 0 から受信した情報を、N F S プロトコル若しくは C I S F プロトコルに従う情報に変換して N F S クライアントや C I F S クライアントに送信する。また、本実施形態では N F S プロトコルと C I F S プロトコルの例を説明しているが、他のプロトコルに対応したゲートウェイを設けることにより、他のプロトコルを用いたクライアントからのファイルシステムアクセスをサポートすることも可能である。

#### 【0058】

以下、本ファイルシステムにおける新規ファイルの作成処理、既存ファイルへの W R I T E 処理および既存ファイルのリード処理を説明する。

#### 【0059】

始めに、図 7 を用いて新規ファイルの作成処理について説明する。

#### 【0060】

クライアント 1 0 0 0 からのファイル作成要求が C N 1 2 0 0 に送信されると、C N 1 2 0 0 のプロセッサは C - S V R 1 2 7 0 を実行する。以下特に断りのない限り、C N 1 2 0 0 のプロセッサが C - S V R 1 2 7 0 を実行することにより行なう処理を、C - S V R 1 2 7 0 が実行すると表現する。

#### 【0061】

クライアントから送信されるファイル作成要求には、作成される新規ファイルが登録される V i e w (即ち作成される新規ファイルの親 View) のハンドルと、作成されるファイルのファイル名およびファイルの操作種別の情報が含まれる。

#### 【0062】

尚、後述の様に、ファイルや V i e w が作成されると、作成されたファイルや V i e w のハンドルがクライアントに通知される。従って、親 V i e w のハンドルは、親 View が作成された際に既にクライアントに通知されているため、クライアントは予め親 V i e w のハンドルを把握しているものとする。

#### 【0063】

また、クライアントは新規ファイルの親 View の View 名を含むオープン要求を、CN に送信し、オープン要求を受信した C - S V R が View 属性エントリ 2 1 0 0 を検索してオープン要求中の View 名に対応する GUID を取得し、更に C - S V R が取得した GUID を用いて O F T から親 View のハンドルを取得し、これをクライアントに通知することとしても良い。この場合クライアントは View 名を管理していれば、オープン要求を CN に送信することにより View のハンドルを取得することができるので、View ハンドルを管理する必要はない。

#### 【0064】

更には、クライアントが親 View の View 名を含む GUID 取得要求を CN に送信し、CN が View 名を用いて View 属性エントリを検索して対応する GUID を取得し、当該 GUID をクライアントに通知することとしても良い。その後、クライアントは通知された GUID を含むファイル作成要求を CN に送信すればよい。この場合、ファイル作成要求中に既に親 View の GUID が含まれているので、後述するステップ 3000 は省略することができ、ファイル作成要求中の GUID を用いて後述のステップ 3010 以降の処理が実行される。

#### 【0065】

C - S V R 1 2 7 0 はファイル作成要求に含まれる親 V i e w のハンドルを用いて O F T 1 2 6 0 を検索し、当該ハンドルが含まれる O F T 内のエントリ (以下ハンドルエントリ) を取得する (ステップ 3 0 0 0) 。

#### 【0066】

次に C - S V R 1 2 7 0 は、取得したハンドルエントリに含まれる G U I D を用いて A T 1 2 4 0 を検索し、当該 G U I D が登録されている V i e w 属性エントリを取得する。ここで、A T 1 2 4 0 の検索は、実際には C - S V R からの検索要求に従って C N 1 2 0 0 が D B M S 1 2 2 0 のプログラムを実行することにより実現する。

#### 【0067】



C-SVR1270はDBMSによって取得したView属性エントリのACL\_MGR\_IDフィールドを参照し、ACLマネージャ1530のIDを取得し、取得したIDを用いてACLマネージャ1530に対してアクセス権限チェック要求を送信する(ステップ3010)。このアクセス権限チェック要求には、ファイル作成要求を送信したユーザのユーザ認証情報(CRED)が含まれる。ACLマネージャは当該CREDで識別されるユーザの、親Viewに対するアクセス権限情報を記憶装置1230D内のデータベースから検索して、当該ユーザが親View内にファイルを作成する権限を有するかどうかチェックする。尚、C-SVRは、上述の様に、ST1210を用いることによってユーザ認証情報(CRED)を取得することができる。

#### 【0068】

次に、C-SVR1270は、ACLマネージャからアクセス権限チェックの結果を受信し、クライアントから指定された親View内へ、当該クライアントが新規にファイルを作成する権限が有るか否かを調べる(ステップ3020)。ファイル作成権限がない場合、C-SVR1270はファイル作成失敗という処理結果をクライアントに返信し(ステップ3080)、ファイル作成処理を終了する。

#### 【0069】

ファイル作成権限がある場合、C-SVR1270は新規ファイルを作成する(ステップ3030)。具体的にはC-SVRは、新規ファイルに対応するファイル属性エントリ2000をAT1240内に設定すると共に、新規ファイルに対してGUIDを割りあて、クライアントから受信したファイル名と共に当該GUIDを新しく設定したファイル属性エントリに格納する。またC-SVRは、クライアントから受信する新規ファイルのファイルデータを、SN1300を介して記憶装置1230B内のファイルコンテナ1310に格納する。そしてC-SVRは、ファイルの作成時刻、ファイルデータのサイズ、ファイルのタイプ、ファイルデータの記憶装置1230B内での格納位置、ACL\_MGR\_ID、ACT\_MGR\_IDも、ファイル属性エントリに登録する。

#### 【0070】

C-SVR1270は、新規ファイルの作成結果を調べ(ステップ3040)、ファイル作成に失敗していることが分かったと、ファイル作成失敗という処理結果をクライアントに返信し(ステップ3080)、ファイル作成処理を終了する。

#### 【0071】

ファイル作成に成功するとC-SVR1270は、新規に作成したファイルの、親ViewのView属性エントリのcontents\_Locとcontents\_infoを用いて、親ViewのViewコンテナから最新のViewデータをCN1200内部のバッファメモリに格納し、バッファメモリ上のViewデータの最後に、ステップ3030で作成したファイルのGUIDと、クライアントから受信したファイル作成要求に含まれていたファイル名を登録する(ステップ3050)。ここでC-SVR1270は、NFSプロトコルまたはCIFSプロトコルのような公知技術を用いてSN1300と通信し、最新のViewデータを取得する。

#### 【0072】

次にC-SVR1270は、バッファ上に作成したViewデータ用のViewデリミタを作成し、記憶装置1230B内に格納されている親ViewのViewコンテナの最後に、SN1300を介して、Viewデリミタ、Viewデータの順でこれらのデータを追記する(ステップ3060)。

#### 【0073】

さらに、C-SVR1270はOFT1260上に、作成した新規ファイル用のハンドルエントリを作成する(ステップ3070)。このときハンドルエントリのHANDLE2300には当該ハンドルエントリのインデックス番号を、GUID2310にはステップ3030で作成された新規ファイルのGUIDを、CRED2320とSESSION2330にはファイル作成要求を受信したセッションに関する情報が格納されているSTのエントリに登録されているCREDおよびSESSIONと同じ値を格納し、親View

のGUID2340には、親ViewのGUIDを格納する。

【0074】

最後に、ステップ3070でC-SVRは、作成したハンドルエントリのインデックス番号であるハンドルと、ファイル作成に成功したという結果をクライアントに返信してファイル作成処理を終了する（ステップ3080）。

【0075】

以上述べたとおり、本発明ではC-SVRがファイルを新規作成する度に、新しいViewデータがViewコンテナに追記され、古いViewデータが上書きされることはないため、任意の時刻においてC-SVRが管理していたファイルのセットを、後に選び出すことができる。

【0076】

尚、図7ではファイルの作成処理について説明したが、新規Viewの作成処理も同様にして実行することができる。Viewを作成する場合には、クライアントは新規Viewが登録される親Viewのハンドル、新規ViewのView名を有するView作成要求をC-SVRに送信する。また、新規Viewの作成の場合には図7のステップ3030の処理がファイル作成処理とは異なり、C-SVRがView属性エントリを設定すると共に、新規ViewにGUIDを割り当て、当該GUIDと新規ViewのView名を対応付けてView属性エントリにする。また、設定されたView属性エントリには、親ViewのGUIDの他、Viewの作成時刻やタイプ等図4に示すView属性エントリ内の他の情報も、C-SVRによって設定される。更にC-SVRはSN1300を介して記憶装置1230B内に新規View用のViewコンテナを作成する。その他は、Viewの作成処理も図7に示すファイル作成処理と同様の処理にて実行することができる。

【0077】

次に、図8を用いてファイル書き込み処理について説明する。クライアントからCN1200に送信されるファイル書き込み要求は、書き込みを行うファイルのハンドル、ライトデータ、書き込み開始オフセット、データサイズで構成される。

【0078】

尚、書き込み対象のファイルのハンドルは、クライアントが予め取得しているものとする。クライアントが当該ファイルのハンドルを取得する手順は、ファイル作成処理においてクライアントが親Viewのハンドルを取得する手順とはほぼ同様である。しかし、Viewとは異なりファイルの場合は、ファイルが更新されると更新分のデータに対し新しいGUIDが割り当てられるので、同じファイル名に対応付けられているGUIDが複数存在する場合がある。

【0079】

そこで、クライアントがファイル名を含むオープン要求をCNに送信して、CNから書き込み対象のファイルのハンドルを取得する場合、オープン要求を受信したC-SVRはクライアントから受信したファイル名を登録している1又は複数のファイル属性エントリのうち、Mdateが最も現在の時刻に近い（言い換えればファイル属性エントリに格納されているGUIDが他のファイルのanc\_GUIDになっていない）ファイル属性エントリを選択して、このファイル属性エントリに登録されているGUID（以下最新オリジナルGUIDと呼ぶ。）を用いてOFTからViewハンドルを取得するものとする。

【0080】

またクライアントがファイル名を含むGUID取得要求をCNに送信した場合には、C-SVRは最新オリジナルGUIDをクライアントに返信するものとする。

【0081】

C-SVR1270は、クライアントからの書き込み要求を受信すると、書き込み要求に含まれるハンドルを用いてOFTを検索し、書き込み対象のファイルのハンドルエントリを取得する（ステップ2600）。

【0082】

次にC-SVR1270は、ハンドルエントリに含まれるGUID（以下オリジナルGUIDと呼ぶ。）を用いてAT1240を検索し、オリジナルGUIDに対応するファイル属性エントリ2000を取得する。ここで、AT1240の検索は、CN1200がDBMS1220のプログラムを実行して実現する。C-SVR1270は取得したファイル属性エントリのACL\_MGR\_IDフィールドを参照してACLマネージャIDを取得し、このACLマネージャIDが示すACL\_MGRに対してアクセス権限チェック要求を送信する（ステップ2610）。このアクセス権限チェック要求には、ファイル作成処理のステップ3010と同様の手順にてC-SVRが取得した、クライアントのユーザ認証情報（CRED）が含まれており、ACL\_MGRはこのCREDを参照してユーザが書き込み対象のファイルに対し、書き込み権限を有するか否かデータベースを検索してチェックする。

#### 【0083】

次に、C-SVR1270は、ACL\_MGRから返信されたアクセス権限チェックの結果を調べる（ステップ2620）。ファイル書き込み権限がない場合、C-SVR1270はファイル書き込み失敗という処理結果をクライアントに返信し（ステップ2670）、ファイル書き込み処理を終了する。

#### 【0084】

ファイル書き込み権限がある場合、C-SVR1270は記憶装置1230B上に新しくファイルコンテナ（以下新規ファイルコンテナと呼ぶ。）を作成するようS-SVR1330に要求し、クライアントから受信したライトデータをS-SVR1330に送信して、S-SVR1330にライトデータを新規ファイルコンテナに格納させる（ステップ2630）。このとき、新規ファイルコンテナの作成、および新規ファイルコンテナへのライトデータの格納処理は、C-SVR1270とS-SVR1330間でNFSやCIFSプロトコルのような公知技術を用いて実現できる。

#### 【0085】

次にC-SVR1270は、新たなGUID（以下新規GUIDと呼ぶ）を新規ファイルコンテナに割り当てると共に、AT1240に新規ファイルコンテナに対するファイル属性エントリ1240（以下、新規ファイル属性エントリと呼ぶ。）を設定して、このファイル属性エントリに新規GUIDを登録する（ステップ2640）。更にC-SVRは、新規ファイル属性エントリのMdateに現在の時刻を、anc\_GUIDにはオリジナルGUIDを、contents\_Locには新規ファイルコンテナの記憶装置1230B内のロケーションを、contents\_infoにはクライアントから受信したファイル書き込み要求に含まれるオフセットとサイズを格納し、新規ファイル属性エントリのname、Cdate、type、ACL\_MGR\_ID、ACT\_MGR\_IDの各フィールドにはオリジナルGUIDが登録されているファイル属性エントリの対応するフィールドと同じ値を格納する。またクライアントから受信したファイル書き込み要求中のオフセットとサイズを足した値である書き込み終了オフセット値がオリジナルGUIDのファイル属性エントリのsizeフィールドの値より大きければ、sizeフィールドには書き込み終了オフセットの値が、小さければsizeフィールドにはオリジナルGUIDのファイル属性エントリのsizeフィールドの値が格納される。

#### 【0086】

次に、C-SVR1270はOFT1260の書き込み対象ファイルのハンドルエントリに登録されている親GUID2340で示されるView（即ち書き込み対象ファイルの親View）に、新規ファイルを登録する（ステップ2650）。この登録処理は以下の手順により実行される。まずC-SVRが、親Viewの最新ViewデータをSNを介して記憶装置1230Bから読み出し、CN1200のバッファに格納する。次にC-SVRは、バッファに格納されたViewデータの中からオリジナルGUIDが登録されているエントリを検索し、当該エントリ中のオリジナルGUIDの値を新規GUIDに変更する。そしてC-SVRは、S-SVRを介して、変更後のViewデータを、最新Viewデータとして親ViewのViewコンテナに追記する。ここで、変更後のView

wデータをViewコンテナに追記する処理は、CN上に読み込んだViewデータ中のオリジナルGUIDを新規GUIDに変更する点以外は、前述したファイル作成処理のステップ3050で説明した処理と同じである。

#### 【0087】

その後、C-SVR1270は、ステップ2600で取得したOFT上のハンドルエントリに登録されているGUID2310の値を、新規GUIDに変更し（ステップ2660）、最後にファイル書き込み処理が成功したことを処理結果としてクライアントに返信し、ファイル書き込み処理を終了する。

#### 【0088】

次に、図12を用いてファイルのリード処理について説明する。クライアントからCN1200に送信されるREAD要求には、リード対象のファイルのハンドル、READ開始オフセット、データサイズが含まれる。尚、クライアントはファイルへの書き込み処理において説明した手順と同様の手順により、リード対象のファイルのハンドルを予め把握しているものとする。

#### 【0089】

C-SVR1270は、クライアントからのREAD要求を受信すると、READ要求に含まれるハンドル番号を用いてOFTを検索し、ハンドルエントリを取得する（ステップ2800）。

#### 【0090】

次にC-SVR1270は、ハンドルエントリに含まれるGUIDを用いてAT1240を検索し、当該GUIDに対応するファイル属性エントリを取得する。ここで、AT1240の検索は、CN1200がDBMS1220のプログラムを実行して実現する。C-SVR1270は取得したファイル属性エントリのACL\_MGR\_IDフィールドを参照して、ACLマネージャIDを取得し、対応するACL\_MGRに対してアクセス権限チェック要求を送信する（ステップ2810）。尚、アクセス権限チェック要求にはユーザ認証情報（CRED）が含まれること、及びACL\_MGRは当該CREDを用いてユーザのファイルリード権限の有無をチェックすること等は、上述のファイル作成処理、ファイル書き込み処理において説明した手順と同様である。

#### 【0091】

次に、C-SVR1270は、ACL\_MGRから返信されたアクセス権限チェックの結果を調べる（ステップ2820）。クライアントを使用しているユーザにREAD対象のファイルに対するREAD権限がない場合、C-SVR1270はファイルREAD失敗という処理結果をクライアントに返信し（ステップ2840）、ファイルREAD処理を終了する。

#### 【0092】

ユーザにファイルREAD権限がある場合、C-SVR1270はステップ2810で取得したファイル属性エントリ中のcontents\_Locを取得し、記憶装置1230B内のcontents\_Locが示す記憶領域上に存在するファイルコンテナから、データを読み出す（ステップ2830）。このとき、contents\_Locが示すファイルコンテナに含まれるファイルデータは、同じくファイル属性エントリ中のcontents\_Infoが示すオフセットとサイズで示される範囲のファイルデータである。このため、クライアントから受信したREAD要求に含まれる開始オフセットおよびデータサイズを、contents\_Infoに含まれるオフセットおよびサイズと比較し、READ要求で示される範囲に含まれるファイルデータのみをcontents\_Locが示すファイルコンテナから読み出す。ここで、READ要求の開始オフセットおよびデータサイズで示される範囲の全ファイルデータ読み出しが完了すれば、ステップ2830を終了する。一方、ステップ2810で取得したファイル属性エントリ中のcontents\_Locが示すファイルコンテナだけで、READ要求で示される範囲のデータ読み出しが完了していない場合、ステップ2810で取得したファイル属性エントリのanc\_GUIDに登録されているGUIDが示すファイル属性エントリ中のcontents\_Locを取得して、当該contents\_Locが示す記憶領域上に存在するファイルコンテナからも前述と同様の方法でデータを読み出

す。ステップ2830におけるデータの読み出し処理は、READ要求で示された範囲のデータ読み出しが完了するか、最後のファイル属性エントリ（即ちanc\_GUIDにGUIDが登録されていないファイル属性エントリ）中のcontents\_Locが示す記憶領域上のファイルコンテナからデータを読み出すまで続けられる。このとき、ファイルデータの読み出し処理は、C-SVR1270とS-SVR1330間でNFSやCIFSプロトコルのような公知技術を用いて実現できる。

#### 【0093】

最後にC-SVR1270は、ステップ2830で読み出したデータ、読み出したデータのサイズ、および読み出し処理が成功したことを示す処理結果をクライアントに返信し、ファイルリード処理を終了する。

#### 【0094】

次に、図9を用いて、Viewデータ取得処理を説明する。クライアント1000は、Viewのハンドルと時刻を指定したViewデータ取得要求をCN1200に送信することによって、指定時刻において指定されたViewに登録されていたファイルやViewを示す、Viewデータを取得することができる。尚、上述のファイル作成処理にて説明した通り、クライアントはViewデータ取得要求の対象となっているViewのハンドルを予め把握しているものとする。

#### 【0095】

C-SVR1270がクライアントからのViewデータ取得要求を受信すると、C-SVRは、Viewデータ取得要求に含まれるハンドル番号を用いてOFTを検索し、指定されたViewのハンドルエントリを取得する（ステップ2700）。

#### 【0096】

次にC-SVR1270は、ハンドルエントリに含まれるGUIDを用いてAT1240を検索し、当該GUIDに対応するView属性エントリを取得する。ここで、AT1240の検索は、CN1200がDBMS1220のプログラムを実行することにより実現される。C-SVR1270はView属性エントリのACL\_MGR\_IDフィールドを参照してACLマネージャIDを取得し、対応するACL\_MGRに対してアクセス権限チェック要求を送信する（ステップ2710）。尚、アクセス権限チェック要求にユーザ認証情報（CRED）が含まれること、C-SVRがCREDを取得する方法、及びACL\_MGRがユーザのViewデータに対するアクセス権限をチェックする方法等はファイル作成処理と同様である。

#### 【0097】

次に、C-SVR1270は、ACLマネージャから返信されたアクセス権限チェックの結果を調べる（ステップ2720）。クライアントを使用しているユーザにViewデータ取得権限がない場合、C-SVR1270はViewデータ取得失敗という処理結果をクライアントに返信し（ステップ2770）、Viewデータ取得処理を終了する。

#### 【0098】

Viewデータ取得権限がある場合、C-SVRはDBMS1220を実行してAT1240を検索し、ステップ2700で取得したハンドルエントリ中のGUIDを含むView属性エントリを取得する。さらにC-SVRは、取得したView属性エントリのcontents\_Locとcontents\_infoを用いて、S-SVR1330を介して記憶装置1230B中のViewコンテナから最新のViewデリミタを取得する（ステップ2730）。

#### 【0099】

次にC-SVRは、取得したViewデリミタ中の時刻データ(date)と、クライアントがViewデータ取得要求中で指定した時刻データを比較する（ステップ2740）。Viewデリミタ中の時刻が、クライアントが指定した時刻よりも古くない場合には、C-SVRはS-SVRを介して記憶装置1230B内のViewコンテナから、一つ前の世代のViewデリミタを取得する（ステップ2750）。C-SVRは、ステップ2740とステップ2750を、クライアントが指定した時刻よりも古いViewデリミタが取

得できるまで繰り返す。

【0100】

C-SVRが、クライアントが指定した時刻よりも古い時刻を持つViewデリミタを取得したら、C-SVRはS-SVRを介してそのViewデリミタに対応するViewデータを読み込み（ステップ2760）、当該Viewデータをクライアントに送信して処理を終了する。

【0101】

クライアントに返されたViewデータには、Viewデータ取得要求においてクライアントが指定した時刻において、Viewデータ取得要求中でクライアントが指定したViewに登録されている、View名とそのView名に対応するGUIDのセット、若しくは、ファイル名とそのファイル名に対応するGUIDのセット、若しくはその双方が含まれている。従って、クライアントは取得したファイルのGUIDを用いてリード要求をCNに発行し、C-SVRはクライアントから受信したGUIDを用いて前述のリード処理を行うことで、クライアントは任意の時点におけるファイルのデータを取得することができる。尚、クライアントはファイルのGUID有するリード要求を発行するので、図12に示すステップ2800は省略され、ステップ2810以降の処理がリード要求中のGUIDを用いてC-SVRによって実行される。

【0102】

この様に、クライアントは最初に時刻とViewを指定したViewデータ取得要求をCNに発行して当該時刻におけるViewデータを取得し、次に取得したViewデータに含まれているGUIDを用いてファイルリード要求を発行することにより、任意時点におけるファイルデータを取得することができる。

【0103】

尚、これら2つの処理（Viewデータの取得とファイルデータのリード）を一括して行うインタフェース（たとえば2つの処理を一括して行うよう要求する特別なコマンド等）を設け、クライアントがファイル名と時刻をCNに対して指定するだけで、任意時刻のファイルデータを取得できるようにすることも可能である。

【0104】

以上に説明した様に、CNはクライアントから受信したデータ各々に異なる識別情報を付与し、この識別情報をクライアントから受信したファイル名と関連付けて記憶装置に格納するので、一旦格納されたデータは更新されることのないよう管理すると共に、ライトデータに付与された識別情報とクライアント計算機から指定されるファイル名とを関連付けて管理する計算機システムが実現できる。

【0105】

また、CNは更新前のファイルデータに付与された識別情報と、ファイルに対するライトデータに付与された識別情報とを関連付けて記憶装置に格納するので、更新前のファイルデータと更新後のファイルデータとを関連付けて管理することのできる計算機システムが実現できる。

【0106】

更に、計算機システムにレガシゲートウェイを設け、NFSクライアントやCIF Sクライアントからの要求をCNが採用しているプロトコルにレガシゲートウェイが変換してCNに送信するので、NFSやCIF Sのように広く普及したファイルアクセスプロトコルを用いているクライアントからもファイルにアクセスすることができる計算機システムが実現できる。

【実施例2】

【0107】

次に図10および図11を用いて、本発明の第二の実施形態を説明する。図10は本発明が適用される計算機システムの他の一例であり、コントロールノード（以下CN）1200とストレージノード（以下SN）1300が複数存在する点、イエローページサーバ（以下Y.P）1900が存在する点、およびクライアント1000とCN1200の対応

を管理するディレクトリサーバ1800が存在する点が図2に示す実施形態と異なる。

#### 【0108】

YP1900は、GUIDが計算機システム内で一意であることを保障するためのサーバである。YP1900は計算機システムの未使用GUIDを保持し、各CN1200からの要求に応じて、一定個数の未使用GUIDを一括してCNに割り当てる。各CN1200は、ファイルやViewが新規に作成された場合や、ファイルが更新された場合に、YP1900から割り当てられた未使用GUIDのうちのひとつ選択して、作成されたファイル、View若しくはファイルの更新データに対して当該GUIDを割り当てる。このように、YP1900が各CN1200に対して未使用GUIDを一括して割り当てることで、システム内におけるGUIDの一意性を確保しつつ、YP1900へのGUIDの問い合わせによる負荷を軽減することができる。

#### 【0109】

ディレクトリサーバ1800は、クライアント1000とCN1200の対応関係を管理するCNテーブル（以下CNT）1810を保持しており、クライアント1000からの問い合わせに応じて、ディレクトリサーバがCNTテーブルを検索して、問い合わせを行ったクライアント1000に対応するCN1200のIDを取得し、このIDをクライアントに返却する。

#### 【0110】

図11はファイルコンテナおよびViewコンテナのロケーションを管理するロケーションテーブル（以下LT）1250の一例を示している。ロケーションテーブルは第一実施例と同様、CNが接続されている記憶装置1230A内に格納されており、CN内のDBMSによって検索される。LT1250は、ファイルまたはViewのGUIDを格納するGUIDフィールド4000と、ファイルコンテナまたはViewコンテナの第一のロケーションを格納するフィールド（以下第一ロケーション）4100と、第二のロケーションを格納するフィールド（以下第二ロケーション）4200を有する。尚図11は、ロケーションテーブルに2つのロケーション情報が登録されている例を示しているが、SNの数および必要な冗長度に応じてロケーションテーブルに登録されるロケーション情報の数を増減させることも可能である。

#### 【0111】

第一の実施形態では、CN1200がひとつしかなかったため全てのクライアント1000およびレガシーゲートウェイ1600からのファイルアクセス要求はひとつのCN1200に送信されていた。本実施形態では、前述の通りクライアント1000とCN1200との対応をディレクトリサーバが管理する。

#### 【0112】

クライアント1000は、ファイルアクセス要求（ファイル作成要求、ファイル書き込み要求、ファイルリード要求、Viewデータ取得要求、オープン要求等を含む。）をCN1200に送信する前に、ファイルアクセス要求を送信するCN1200をディレクトリサーバ1800に問い合わせる。この際クライアントはディレクトリサーバに自身のIDを通知する。ディレクトリサーバ1800はクライアント1000からの問い合わせを受信すると、問い合わせを送信してきたクライアントのIDを用いてCNT1810を検索し、一致したクライアントIDに対応するCN1200のIDを取得する。CN1200のIDが取得できると、問い合わせ要求を送信してきたクライアントに対してCN1200のIDを送信する。

#### 【0113】

本実施形態では、CNT1810におけるクライアントとCN1200との対応は、クライアント1000の数とCN1200の数を勘案し静的にCNTに登録されている。しかし、一つのCN1200に対して同時にアクセスしているクライアント数に偏りが発生した場合、CNTに登録されているクライアントとCNとの対応関係を動的に変更することも可能である。その場合、クライアント1000からの問い合わせ要求とそれに対する返信をディレクトリサーバ1800内で管理しておき、各CN1200に現在割り当てら

れているクライアント 1000 の数を管理しておく必要がある。

【0114】

ディレクトリサーバ 1800 からのアクセスすべき CN の ID を受信したクライアント 1000 は、受信した CN 1200 の ID を用いてファイルアクセス要求を送信する。この後は第一の実施形態で記述した処理と同様の処理を行なう。

【0115】

このように、複数の CN 1200 を計算機システム内に設けることにより、クライアントからのアクセスによる CN の負荷を適切に分散することができ、計算機システム全体の性能が向上する。また、片方の CN 1200 に障害が発生した場合でも、ディレクトリサーバがそれを検知し、障害検知後のクライアント 1000 からの問い合わせ要求に対して障害が発生した CN 1200 以外の CN の ID を返すようにすることで、クライアントからのアクセス要求を処理し続けることが可能となり、計算機システムの可用性が向上する。

【0116】

更に、第一の実施形態においては、ファイルコンテナまたは View コンテナを格納する SN はひとつのみであったが、本実施形態においては、ひとつのファイルあるいは View に対して複数のファイルコンテナあるいは View コンテナを割り当て、異なる複数の SN を介して異なる複数の記憶装置に格納することができる。

【0117】

第二の実施形態においては、C-SVR 1270 がファイルコンテナあるいは View コンテナをアクセスする際、まず C-SVR 1270 が DBMS 1220 を起動して記憶装置内の LT 1250 をアクセスし、第一ロケーション 4100 もしくは第二ロケーションのいずれかを選択するようにする。例えば、GUID の値に応じて C-SVR が選択するロケーションを第一ロケーションか第二ロケーションのいずれかにあらかじめ決めておく。そして、C-SVR 1270 は選択したロケーションが示す S-SVR に対してファイルコンテナ若しくは View コンテナのアクセス処理を実行する。この結果 SN 間での負荷分散が可能となる。また、片方の SN に障害が発生した場合、障害の発生を CN 1200 が検知し、障害検知後にファイルコンテナ若しくは View コンテナにアクセスする場合には障害の発生していない SN を選択することで、計算機システムの耐障害性を向上させることができる。

【図面の簡単な説明】

【0118】

- 【図 1】 ファイルシステムの概念の一例を示す図である。
- 【図 2】 計算機システムの一例を示す図である。
- 【図 3】 ファイルの構成の一例を示す図である。
- 【図 4】 View の構成の一例を示す図である。
- 【図 5】 オープンファイルテーブルの一例を示す図である。
- 【図 6】 セッションテーブルの一例を示す図である。
- 【図 7】 新規ファイル作成処理の一例を示す図である。
- 【図 8】 ファイル書き込み処理の一例を示す図である。
- 【図 9】 View データ取得処理の一例を示す図である。
- 【図 10】 計算機システムの他の一例を示す図である。
- 【図 11】 ロケーションテーブルの一例を示す図である。
- 【図 12】 ファイルのリード処理の一例を示す図である。
- 【図 13】 SN 及び記憶装置の一構成例を示す図である。

【符号の説明】

【0119】

1000…クライアント  
1200…CN  
1300…SN

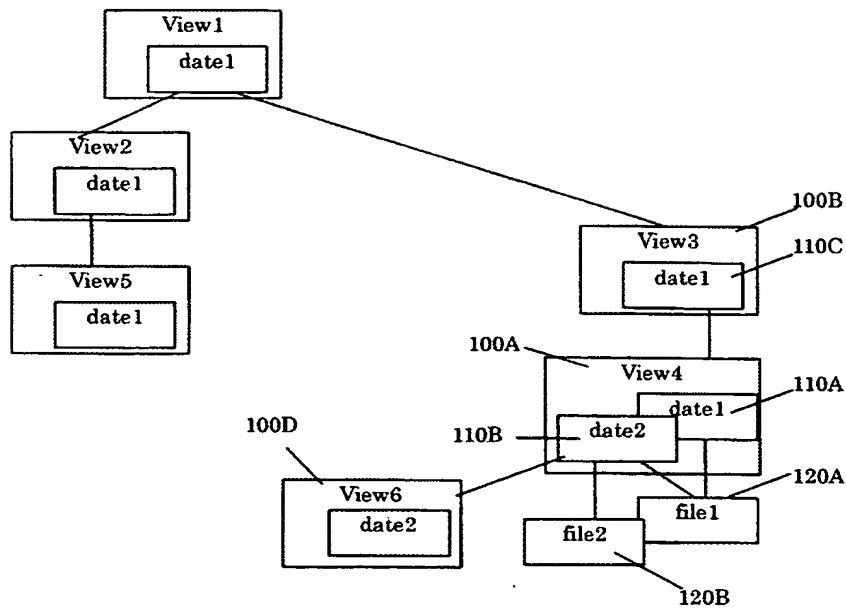


1 2 3 0 …記憶装置  
1 7 0 0 …ネットワーク  
1 6 0 0 …レガシーゲートウェイ  
1 4 0 0 …管理ノード  
1 5 0 0 …外部ノード

【書類名】 図面

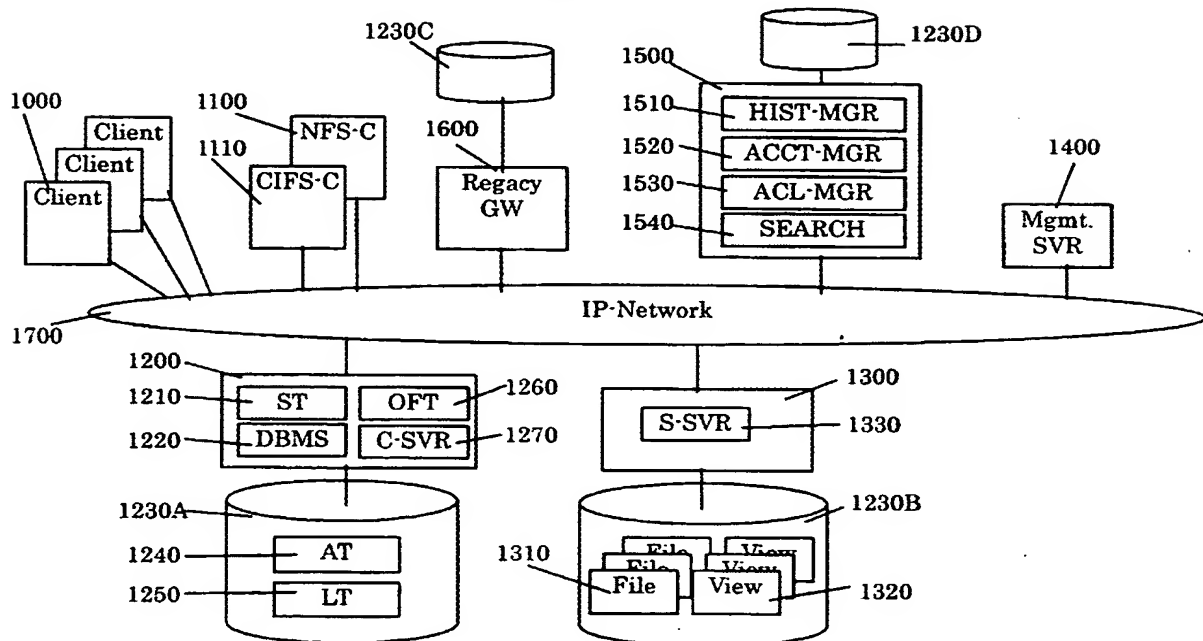
【図 1】

図 1



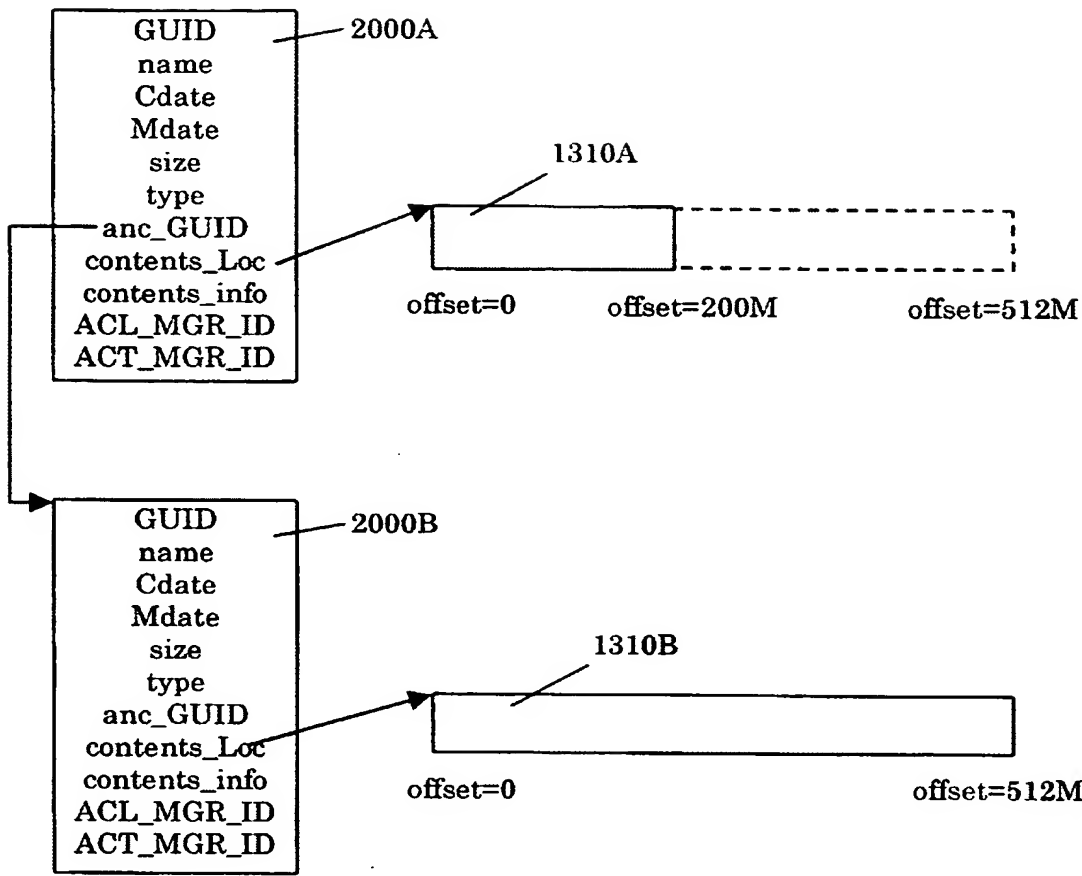
【図 2】

図 2



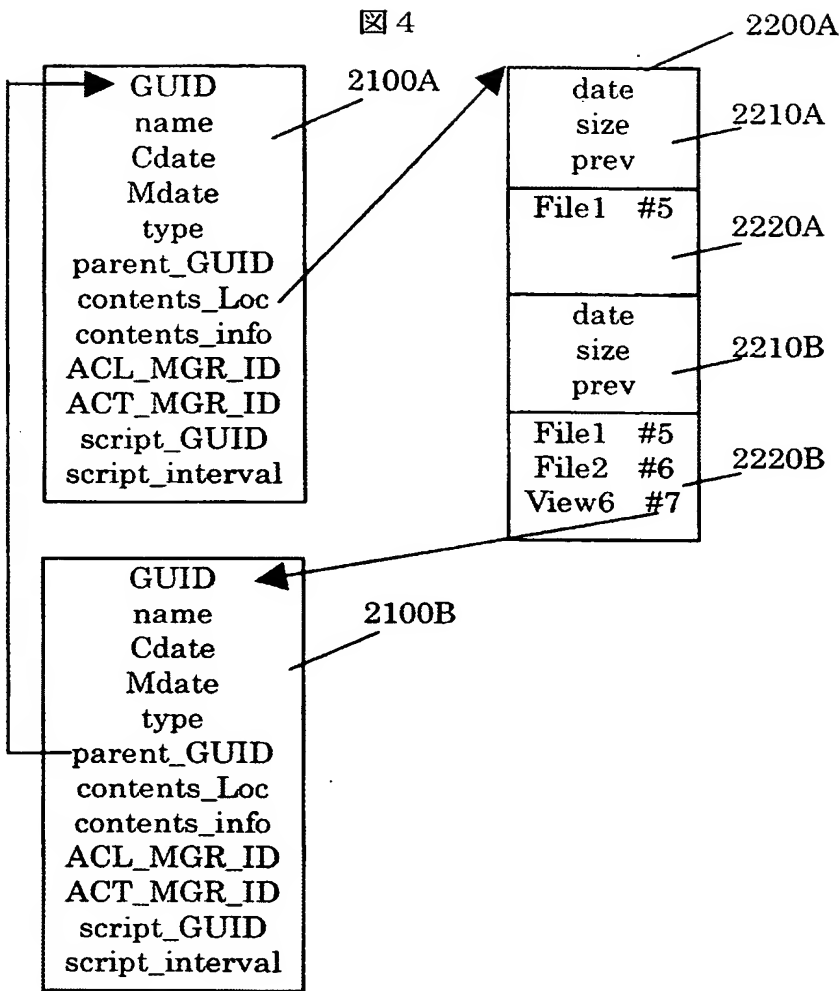
【図 3】

図 3



ファイルの構造

【図 4】



View の構造

【図 5】

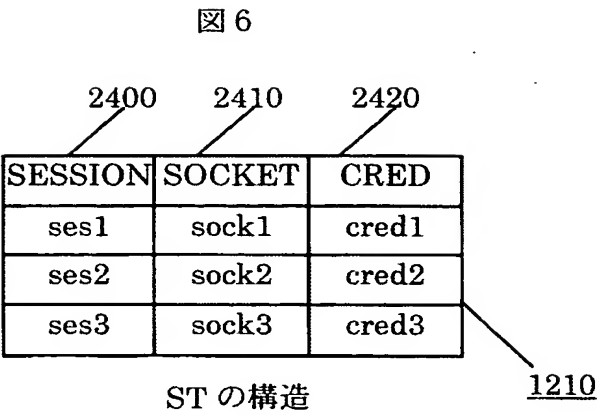
図 5

2300	2310	2320	2330	2340
HANDLE	GUID	CRED	SESSION	PARENT
h1	guid1	cred1	ses1	p1
h2	guid2	cred2	ses1	p1
h3	guid3	cred3	ses1	p2

1260

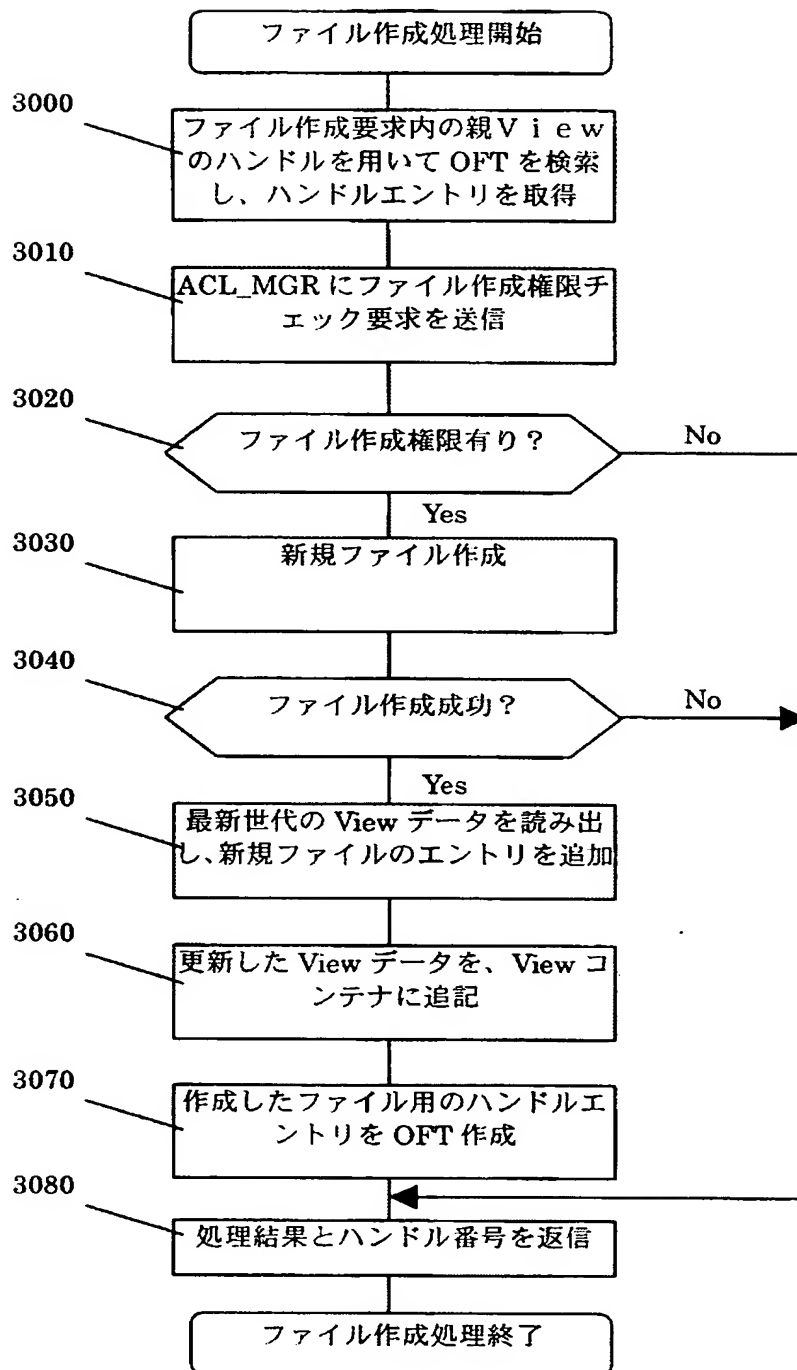
OFT の構造

【図 6】



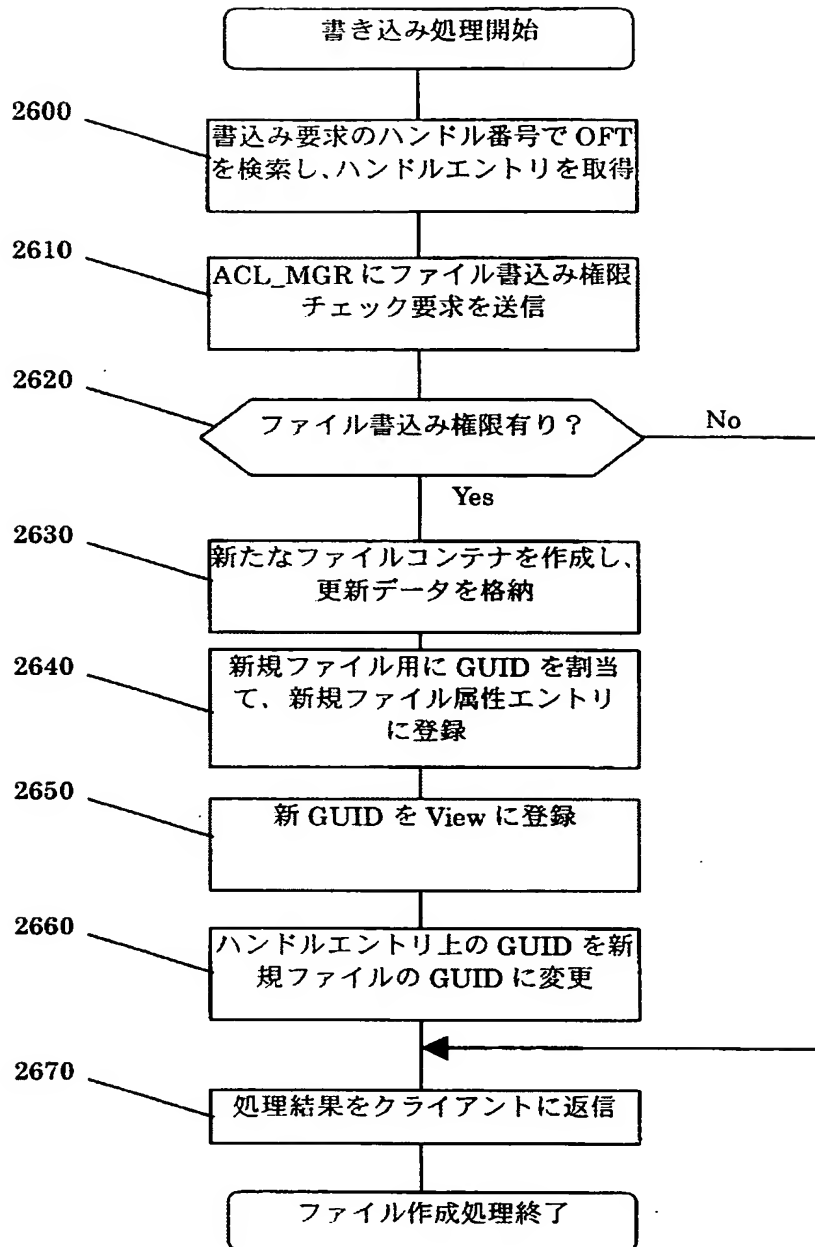
【図 7】

図 7



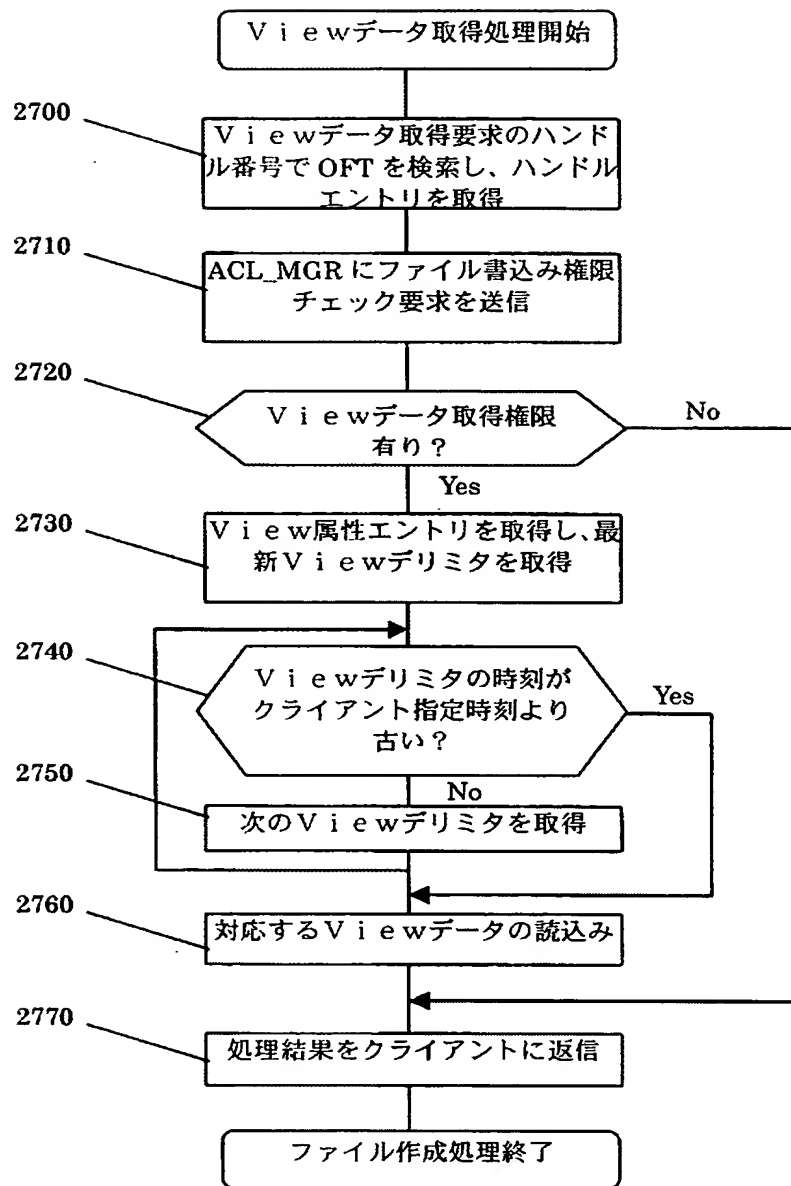
【図 8】

図 8



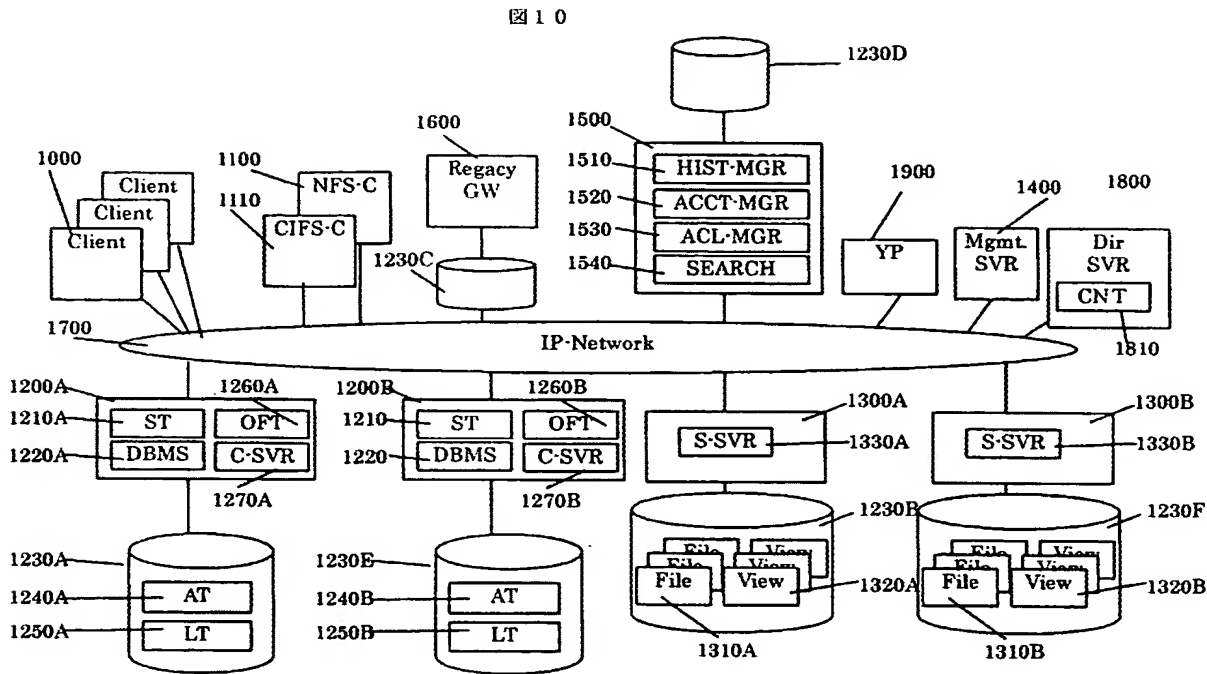
【図 9】

図 9





【図 1 0】



【図 1 1】

図 1 1

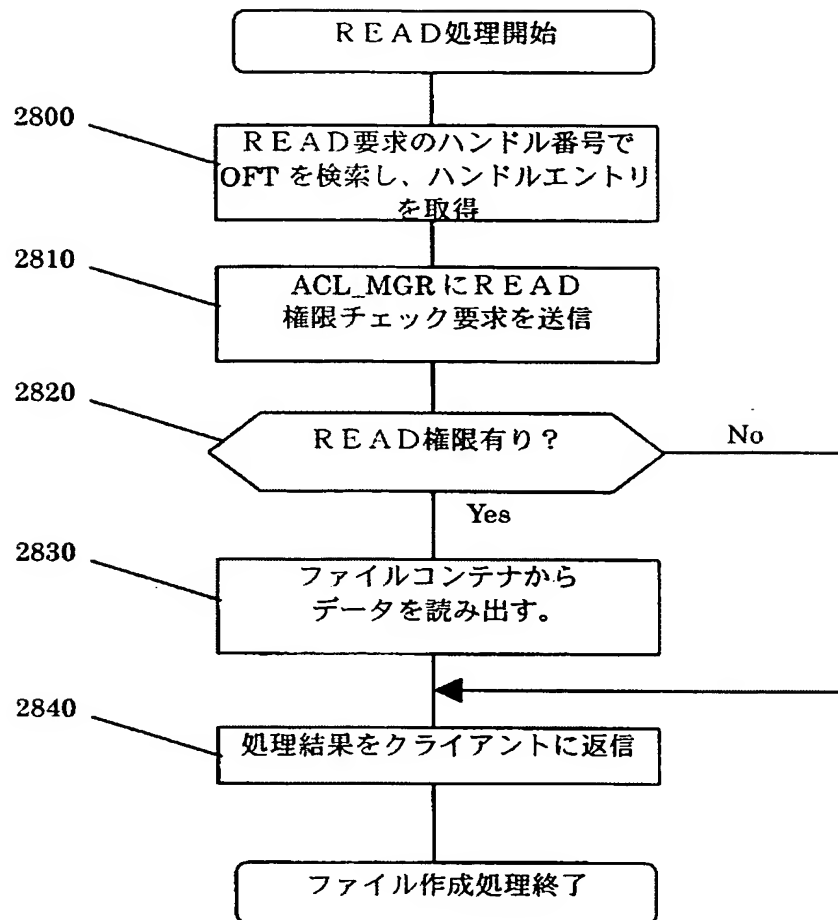
1250

4000		
4100		
4200		
GUID	Location 1	Location 2
guid1	SSVR1:/FILE1	SSVR2:/FILE1
guid2	SSVR1:/FILE2	SSVR2:/FILE2
guid3	SSVR1:/FILE3	SSVR2:/FILE3

LT の構成

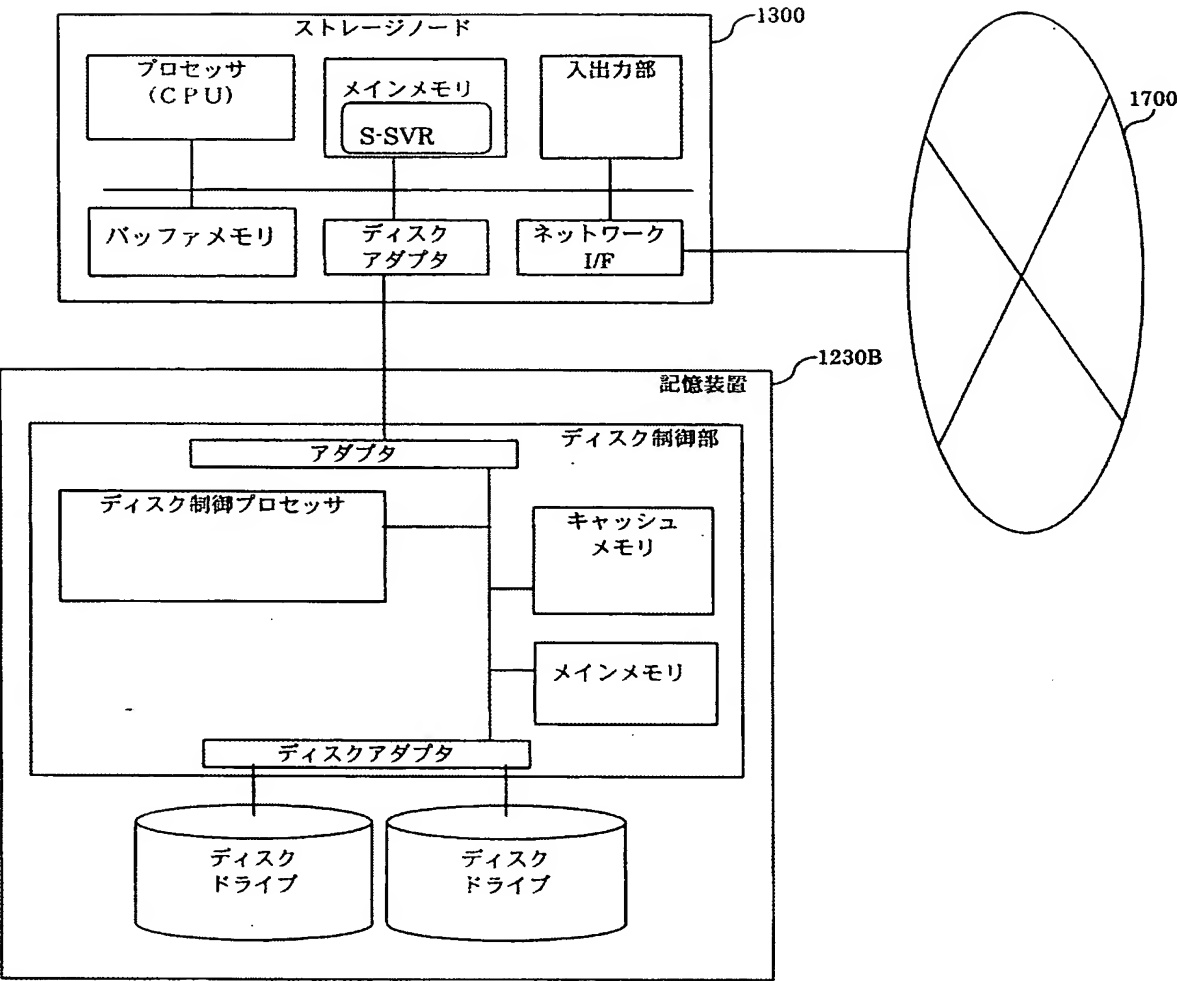
【図 12】

図 12



【図 13】

図 13



**【書類名】 要約書****【要約】****【課題】**

従来の計算機システムでは、ファイルのデータが更新されないことを保障しつつクライアントから指定されたファイル名を用いてファイルを管理することができなかった。

**【解決手段】**

計算機システムは、クライアントからファイルへのアクセス要求を受信する第一の計算機と、第一の計算機に接続され、ファイルの管理情報を格納する第一の記憶装置と、第一の計算機からデータへのアクセス要求を受信する第二の計算機と、第二の計算機に接続され、ファイルデータを格納する第二の記憶装置とを有する。

第一の計算機は、クライアントからファイルデータを受信する毎に、ファイルデータに識別情報を割り当てて、クライアントから指定されるファイル名と当該識別情報とを第一の記憶装置に格納する。また第一の計算機はファイルデータを第二の計算機を介して第二の記憶装置システムに格納する。

**【選択図】 図 1**

## 認定・付加情報

特許出願の番号	特願 2 0 0 3 - 3 8 3 2 4 9
受付番号	5 0 3 0 1 8 7 4 8 0 2
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 1 1 月 1 4 日

## &lt; 認定情報・付加情報 &gt;

【提出日】	平成15年11月13日
-------	-------------

特願 2 0 0 3 - 3 8 3 2 4 9

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 5 1 0 8 ]

1. 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台 4 丁目 6 番地

氏 名

株式会社日立製作所